View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Springer - Publisher Connector

IT-Implementation of Subject-Oriented Business Processes

10

10.1 To Go

Now, I am excited! So far, everything went more or less without IT. But is it not necessary to let the process run on my systems?

I am extremely keen on mapping the process on IT systems. I see high potential for motivation and optimization. In addition, IT should guide our staff's work and support them in doing their job.

The natural language sentence semantics with subject, predicate, and object can be transferred to concepts of IT—and quite seamlessly! There is no need for employees to deal with unknown terrain; they can immediately just 'do'.

Natural language sentence semantics allows us to make use of approaches, such as service-oriented architectures, in a straightforward way. In the realm of S-BPM, SOA even takes on a new meaning. Services are ultimately the predicates a subject uses.

A. Fleischmann et al., *Subject-Oriented Business Process Management*,
DOI 10.1007/978-3-642-32392-8_10, © The Author(s) 2012

IT has achieved a high level of penetration in many organizations. Without IT support, many business processes cannot be handled in an economically beneficial way. For this reason, the careful and on-demand mapping of processes to information and communication technology is an important task. This applies for cases where employees are involved, as well as for operations in which a high degree of automation is striven for. A suitable and well-fitting software environment plays a significant role here. However, the challenge in many cases is an existing heterogeneous landscape of systems and services, in which each of the components fulfills specific tasks, and for which all of these components need to be integrated into an overall solution for adequate process support.

In this chapter, we first describe the roles of S-BPM stakeholders in the IT implementation (Sect. 10.2). Then we introduce a framework for IT implementation of subject-oriented process models (Sect. 10.3) and describe the IT implementation of subjects and their behavior (Sects. 10.4 and 10.5). Finally, we show that service orchestration is not only an effective but also efficient way to support the dynamics of S-BPM (Sect. 10.6).

10.2 S-BPM Stakeholders in IT Implementation

10.2.1 Governors

The IT manager (e.g., CIO) plays a superior Governor role in IT implementation. He calls for IT compliance of planning, development, and operation of IT solutions (see Sect. 3.6.4). This ranges from the fulfillment of legal requirements (e.g., data protection, principles of data access, and verifiability of digital documents) to the observance of standards and internal guidelines, which the organization itself has defined as binding (e.g., IT infrastructure library, IT architecture principles, IT security policies, etc.). In large organizations, particular roles need to be installed, such as IT security and data protection officer, which will also take over functions of Governors and need to be involved in the IT implementation of processes. This also applies to staff representative bodies such as the works council, which can exert Governor functions, as a result of codetermination regulations.

An important task in the Governor's activity bundle of IT deployment is the process-related assignment of permissions to subjects or subject carriers to enable access to functions and data in the solution. In these cases, the process owner can be Governor. The implementation will be performed by a system administrator in the role of an Actor.

10.2.2 Actors

Actors involved in the process represent the users of solutions for process support. As such, they play an important role in IT implementation. Their behavior specified in the model defines the functional requirements for the systems to be developed.

The Actors can be involved at an early stage of IT implementation by participating in the design of user interfaces and functionality. They can also try out prototypes. They test solutions using specific test cases and data, which they themselves have designed, eventually assisted by Experts.

With the help of Enterprise Mashups, process participants may step increasingly into the role of producing small applications to support their tasks in the process. Prerequisites are an Enterprise Mashup platform with which users can orchestrate information and application services without programming, as well as governance rules, which control and monitor these activities (Pahlke et al. 2010, pp. 302 and 307). This type of end-user computing is particularly suited for situation-specific processes with individual needs and workflows and can be "understood as the next step toward a distributed workflow management by knowledge workers" (Pahlke et al. 2010, p. 307). Given these properties, Enterprise Mashups can serve on the IT technical side as catalyst for self-organization in S-BPM.

10.2.3 Experts

Typical Experts in this bundle of activities are IT professionals, such as IT architects, software developers, database specialists, hardware specialists, and system administrators. They support the Governors, Facilitators, and Actors in building the IT infrastructure for process execution.

10.2.4 Facilitators

A key Facilitator for IT implementation is the leader of a development project. He coordinates the implementation of the domain model into a workflow and all associated tasks. In the development process, he integrates the Actors according to their demands and suggestions, the Governors with respect to their constraints, the Experts with their know-how, and if required, external resources for specific tasks (e.g., training providers).

After going into production with a solution, the responsibility for maintenance and further development is usually passed on to the IT (line) unit. It could also be outsourced to an external service provider. In both cases, troubleshooting and change requests are usually handled by a service desk. Its employees will act as Facilitators, receiving requests for small maintenance tasks or changes during operation. For major modification proposals, they address the respective process owner, who in this case initiates as a Facilitator a (change) project when appropriate.

10.3 Framework for Executing Subject-Oriented Processes

To implement IT support, a business process needs to be represented as a workflow. This is the detailed specification of a process from an IT perspective (cf. Vogler 2006, p. 40). From several conventional interpretations of existing workflow definitions (see, e.g., Becker et al. 2008, p. 56; Gadatsch 2010, pp. 46 ff.; Schmelzer et al. 2010, p. 420; WfMC 1997, p. 244), the following understanding of a workflow can be derived:

A workflow is a:

- Formal description of
- Activities which are executed by
- Communicating actors (roles/people, embedded IT systems)
- Partially or fully automated on
- Objects (inputs and outputs, including data structures)
- Following business rules
- Controlled by the business logic

A workflow is a refinement of a purely domain-specific business process with respect to implementing a strategy (what?) in terms of IT support (how?) (cf. Gadatsch 2010, p. 53).

Referring to the concepts presented in Chap. 5 concerning subject-oriented modeling, and putting these into relation with essential workflow characteristics, the relationships depicted in Fig. 10.1 can be complemented in the right column with the corresponding aspects with respect to IT implementation.

Workflow characteristic	Illustration in the subject-oriented approach	Aspect of IT implementation
Actors (roles/persons)	* Subjects/subject class * Subject carrier (result of the organizational implementation)	Implementation of subjects and subject carriers via the user management of systems (e.g., via LDAP, role and rights concepts, etc.)
Actors (IT system)		Implementation of the action behavior of mechanical subjects (integration of existing and new applications, e.g., as Web Services)
Activities controlled by business logic and rules	* Subject behavior (internal strictly sequential ⇒ orchestration) * Communication structure (subject interactions ⇒ choreography)	Implementation of the action and communication behavior of subjects as steps in Workflow Engine / Business Rule Engine with the integration of existing and new applications [e.g., as Web services]
Communication of the actors	Communication structure (subject interactions), including synchronization of message exchanges using the input pool	Implementation of the communication behavior of the subjects, e.g., by e-mail, etc.
Objects (inputs and outputs incl. data structures)	Business objects	Implementation of business objects and their manipulation in the action behavior of the subjects by user interaction (screen forms) or automated application functionality (e.g., program-triggered database transactions)

Fig. 10.1 Workflow characteristics equivalent in the subject-oriented approach and related aspects of IT implementation

Whereas for organization-specific implementation, the relation of process models to the organization, including underlying human actors (subject carriers) was discussed (see Fig. 10.2, upper part), in the context of IT implementation, the focus is placed on the relation of a process model to IT systems (Fig. 10.2, bottom

part). In the course of IT implementation, also the assignment of subject carriers to subjects needs to be done according to the result of the previously performed organization-specific implementation.

Figure 10.2 shows the frame of reference (framework) integrating humans and machines in a socio-technical system for process execution. As revealed by the figure, models of business processes couple human actors with supporting IT solutions, while they control the process. If the formal model description is transformed into an interpretable language for a workflow engine, the engine can take over the control flow at runtime. It triggers people and application systems as actors according to the workflow specification, supports their individual activities and their cooperation by providing guidelines, information, etc. and documents the progress of processing.

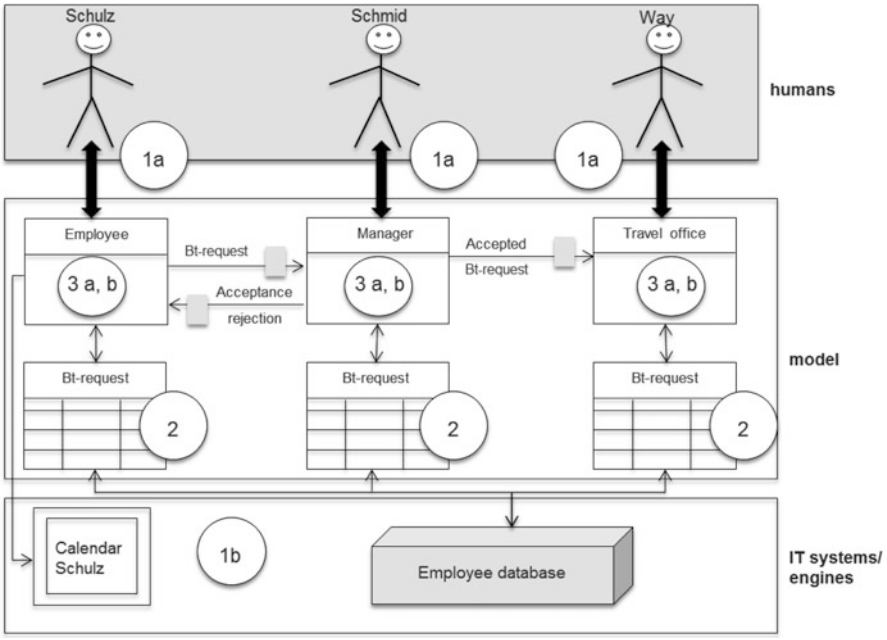

Fig. 10.2 Framework of IT implementation for a subject-oriented process model

In this context, the principle of constructing models systematically becomes essential: taking into account technical systems, such as information systems, the data and functions perspective is, in addition to the stakeholders perspective, in the focus. For IT implementation, the organization-specific implementation needs to be explored and specified in terms of data management, service architecture, and user privileges, and implemented accordingly.

The circled numbers in Fig. 10.2 represent the following aspects of an IT implementation, which are detailed in the following sections:

1. Implementation of access for subject carriers
   (a) Humans (roles/people)
   (b) IT systems/machines
2. Implementation of business objects (see 3a)
3. Implementation of subject behavior (business logic and business rules)
   (a) Behavioral action (manipulation of business objects)
   (b) Communication behavior (sending and receiving messages)

10.4 IT Implementation of Subject Carrier Access

Subjects were assigned to subject carriers performing concrete actions during the organization-specific implementation. In terms of IT implementation, these can be human subject carriers (people as users) or automated subject carriers (IT systems).

Human Subject Carriers

People who are engaged as subject carriers in activities in an IT environment for workflow support must be made known to this environment as users and provided with the required access privileges.

These privileges can be static, but can also change dynamically depending on the organizational context and the progress when executing process instances. For example, the employees of the travel office should only have access to personal data provided by applicants, as long as they work on the travel request. A short-term designated delegate must have the same system and data access privileges as the subject carrier who delegated him.

The implementers could realize user and privilege administration either specifically in the individual applications, or with the help of overall user access concepts, e.g., using the Lightweight Directory Access Protocol (LDAP). A single sign-on should be provided, as actors may need to use many different applications for task completion.

Automated Subject Carriers

For organizational implementation, we have shown how subjects are mapped to human carriers. For IT implementation, subjects need to be assigned to automated subject carriers. IT systems acting in such a process must be integrated into the workflow. To accomplish this tasks, interfaces need to be created which enable the communication between automated subject carriers and also between automated and human subject carriers. Automated subject carriers are mainly used for parts of workflows that can run with minimal human intervention.

> Workflow Management Systems facilitate the straightforward implementation of those parts of subject behavior specifications that can be executed without human intervention. In S-BPM, the subject behavior specification reveals the stakeholder intervention and control requirements for task accomplishment.

10.5 IT Implementation of Subject Behavior

The modeled behavior describes the action behavior (work steps) and the communication behavior (sending and receiving) of the subjects involved in the process (see Sects. 5.5.5. and 5.5.3). The type and sequence of activities of the model determine the business logic of the process which is to be implemented.

The implementation has to create a process flow control and to integrate applications and services providing the functionality required for performing work and interaction steps. For the implementation of the process flow control, the developers may use standardized technologies, such as Java and Business Process Execution Language (BPEL) in conjunction with a workflow engine. Services can be integrated by linking, as a portlet, by calling methods, or as Web services. In this way, when required, the human users can also become part of workflows, e.g., by triggering a service to display a user interface enabling users to enter data into a business object.

The following sections detail various IT implementations of action and communication behavior, exemplifying its use.

10.5.1 Action Behavior

Action behavior includes internal functions a subject or its respective carrier executes in the course of processing a process instance. Of particular importance are operations on business objects. Business objects and possible operations on business objects, or respectively, on their instances, were introduced in the context of modeling (see Sect. 5.5.7.6). The business objects defined in a process model are transformed in the course of IT implementation into appropriate data structures that can be processed by IT systems (e.g., XML schemata).

In a further step, operations on business objects need to be implemented. Figure 10.3 shows various approaches. They are usually applied in combinations.

Subjects that perform operations as part of their behavior for creating and manipulating business objects and business object instances (as shown in the figure) can be users (human subject carriers) or applications (automated subject carriers). They require functions for creating, viewing, editing, storing, etc. of business object content.

10.5.1.1 Human Operators

If users should interactively perform operations on business objects and their instances, they need user interfaces. These can be provided either by an application managing the business object or generated from the data structure description of the business object.

- *Using the user interface (front end) of an application (IT system).* The behavioral description of a subject can define in a state that a subject carrier uses a particular application to modify business object data. For this purpose, the application's

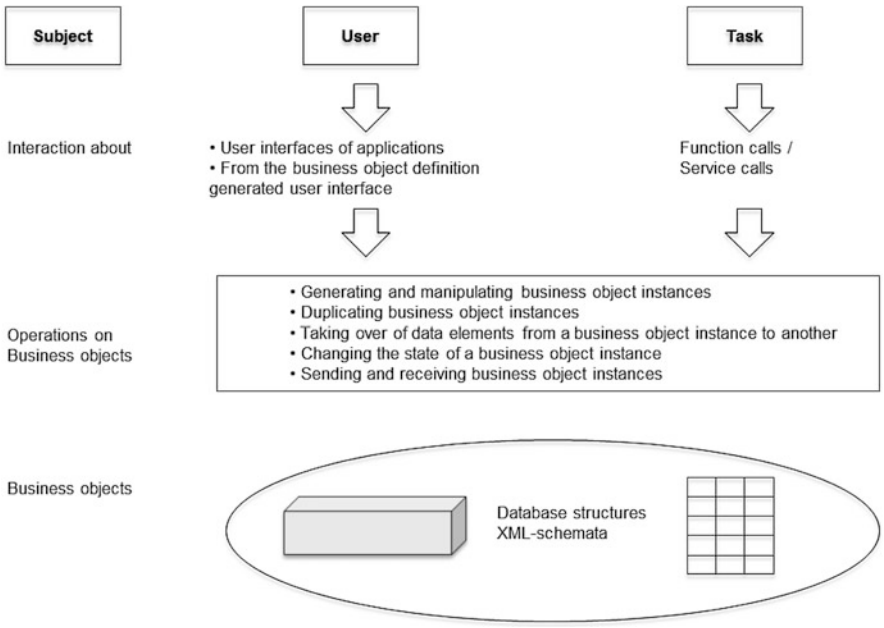

Fig. 10.3 IT implementation of operations on business objects and their instances

screens are directly activated in this state and entered data stored by way of the same. An example of this type of integration of business objects are SAP application transactions. The behavioral description controls the invocation of the transaction, which is represented and implemented in the process model by the abstract business object. In this case, the application can be considered as an encapsulated business object in which the data structure and the user interface are unified.

The technical implementation for the integration of user interfaces of such encapsulated business objects is very straightforward. The transfer of their data into other business objects and vice versa, however, is generally more complicated. This is due to the fact that the complete data structure of an encapsulated business object often remains to a great extent hidden, and only those elements displayed in the associated user interface are visible. Consequently, if elements of the business object need to be accessed without involving the user interface, transfer programs need to be developed to transfer the desired data from the encapsulated object, e.g., from an SAP database, to a target business object, and vice versa.

- *Generating the user interface from the business object definition*. For the manipulation of business objects without recourse to existing applications, the user interface can be derived from the data structure description of the business object. The elements of the business object are mapped to corresponding fields of a screen mask. In case the behavior description contains user interactions, the

subject carrier is able to maintain the data by means of this screen mask. The newly entered or modified values are stored in the corresponding data elements of the business object definition.
If the implementation of the user interface is restricted to simple, table-like user dialogs, its code could automatically be generated from the business object definition using appropriate technology, e.g., http://www.ecplise.org. This also applies for static validation checks for preventing input errors. For instance, in a field that is defined as a date field, only data in a valid date format can be entered; arbitrary strings are not permitted. For a field where only certain inputs are allowed, a bulleted list of the possible values can be defined (value range).

More sophisticated designs leading to more comfortable user interfaces can be achieved through usage of dedicated design tools for user interface screens and forms. However, usually a manual mapping of data elements of the business object to (form) fields is required. Complex, dynamic plausibility checks also require more effort, e.g., due to the need for programming special tests. An example is the dependency of an input on previously entered data. For instance, an underage trainee, after entering his date of birth in the business trip application form, might subsequently be required to enter his legal guardian's data in a dynamically displayed input field.

10.5.1.2 Operations Through Application Functions or Services

Instead of being operated interactively, business objects or their instances can be manipulated automatically and without user intervention by application program functions or services. For implementation, internal functions of a subject behavior are linked to appropriate application functions or services. The flow control component of the workflow engine then invokes these when a subject carrier reaches the respective functional state.

Such functions or services could be database queries or calculation algorithms. They are forwarded business objects to be manipulated, or parts of them, as parameters. They then return results from querying and calculation, respectively, which are transferred to the business object data. The reverse path, e.g., updating data base records from a business object, can also be performed.

In the example of the business trip application, a service could automatically be triggered after an employee has entered the business trip data, in order to calculate advance payments. This service receives a part of the business object "business trip request" with the relevant data for determining the advance payment for the trip, passed on as parameters (e.g., employee number, start and end date, national/foreign country, salary grade, amount of the advance payment [empty], etc.). Using this data, first the service accesses a database in which the expense rates are structured according to destinations and salary groups. Then, it calculates the amount according to the duration of the trip. The calculated value flows back as parameter into the appropriate field of the specific instance of the business object "business trip request".

10.5.2 Communication Behavior

Subjects interact and synchronize by exchanging messages, which often contain business objects. As described in the context of modeling, the concept of input pool is used for implementation (see Sect. 5.5.5.2). Each subject must have such an input pool. IT managers may implement a pool as parameterized service module (e.g., using Web services). It provides insertion and extraction operations and associated interfaces with which subject carriers can deposit outgoing messages and extract received messages.

The extract interface is a local internal affair of the subject and can be implemented by any technology. As a subject usually communicates with several other subjects, however, for the realization of outgoing messages, it should be noted that for sending messages to different recipients different technologies may have to be used (such as Remote Method Invocation (RMI) and Web services). If these are known, in the course of generating code for the subject behavior, the appropriate send operation can be embedded.

When sending a message that contains a business object, only a copy of the business object is created and sent. When receiving a message, the values are taken from the received business object and put into a uniform business object of the receiver. The implementation of these operations can be part of code generation for the behavior of a subject.

10.5.3 Example

The scenario in Fig. 10.4, namely registration and approval of a business trip request, illustrates the combination of the presented possibilities for manipulating business objects, which is often required in practice, as well as the communication of the involved subjects.

The subject “employee” has been linked in the course of the organizational implementation to Mr. Schulz as subject carrier. In the state “complete business trip request”, he fills out an instance of the business object “business trip request”. In order to complete this task, he uses the automatically generated screen mask (from the business object definition), and initially enters his personnel number into the respective entry field. In the background, a function (database query) checks automatically whether for this personnel number, forwarded as a parameter, a record in the employee database exists. It returns either an error message, or data of the person, such as name, first name, salary grade, etc., which are incorporated into the appropriate fields of the business object instance.

For entering the trip start and end date, the electronic calendar of Mr. Schulz is integrated (with a specific) user interface as encapsulated business object. The clicked dates are forwarded by an operation right from the calendar to the business object.

Further information from Mr. Schulz, with respect to destination and the intention of the trip, completes the application instance of the business trip request,

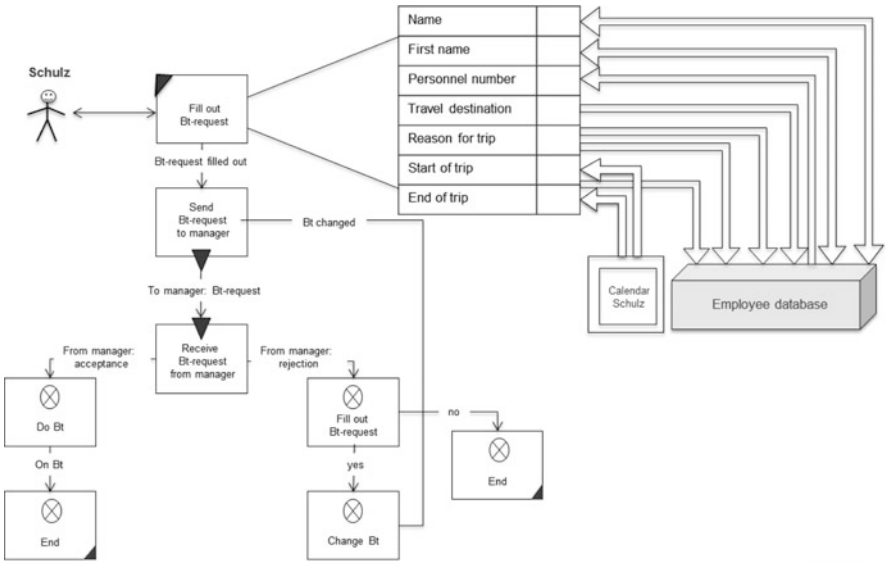

Fig. 10.4 Embedding of the subject employee in the organization-specific and IT environment

which is then sent for approval to Mr. Schmid, the organization-specific implemented manger of Mr. Schulz.

Mr. Schmid sees the arrival of the application process in his process portal and opens it. The data entered by Mr. Schulz and the automatically generated data of the request are enriched for the manager with the notice for approval (e.g., a checkbox with a remark field) and shown on the screen. Mr. Schmid approves the trip without any changes, clicks the appropriate check box, and executes thereby a state transition in accordance with his modeled behavior, namely from the function state "Business trip application—check request" behavior to the send state "Approve".

With this state transition, not only the delivery of the approved application to the applicant is achieved. The approval is also the trigger for the automatic update of a number of databases. A function call linked with the state transition results in the transfer of selected data from the business trip request (travel time, target, intention, etc.) into the employee database. Another function transmits the approved advance payment to the payroll system, which initiates the payment. At the same time, with a corresponding call, the flextime application is triggered to take over the travel dates of Mr. Schulz, which were transferred as parameters from the business object and store them in its own database including presence and absence times, working time balances, etc.

10.6 Relationship to Service-Oriented Architectures

With the use of existing and newly developed applications and services within subjects, the subject-oriented approach forms a solid foundation for building service-oriented architectures (SOA). This architectural principle for software systems provides for the representation of business logic a loosely coupling of largely independent function modules with clearly defined functional tasks (services) (cf. e.g., Krcmar 2010, pp. 345 and 494; Reinheimer et al. 2007, pp. 7).

> Service-oriented architectures allow the implementation of the functional part of subject models in a straightforward way. In S-BPM, all functions of a subject, which are linked to calls of application systems, are affected.

Subject orientation combines the two SOA management concepts of orchestration and choreography as needed (cf. Decker et al., 2007, p. 296). The strictly sequential services for the realization of the subject behavior are orchestrated. The synchronization of the parallel activities of multiple subjects with messages, possibly even across organizational boundaries, corresponds to the principle of choreography. Consequently, subjects of a process can be implemented and run on different IT platforms or workflow engines, respectively. Only the communication between them must be standardized, e.g., via an appropriate Web service agreed upon between all affected parties.

> The principle of coordination in S-BPM corresponds to the same in choreography. In contrast to orchestration, the coordination of subject behavior is achieved by direct message exchange, which simultaneously represents the control of the entire system, and as such, the organization.

Especially in historically grown, heterogeneous, and complex IT environments that are typical for many organizations, the approach thus helps to achieve the goals of SOA. These aim to make software systems more flexible and to adapt them more easily and more quickly to changing operational requirements, particularly at the level of business processes (Reinheimer et al. 2007, pp. 7 et seq.)

10.6.1 Services in Subject Orientation

In the previous sections, it was shown that subjects use services in their behavior to perform operations on business objects and to exchange messages. These services can be of different nature:

- On one hand, they can be function blocks, already developed following the principles of service orientation, which have characteristic features such as

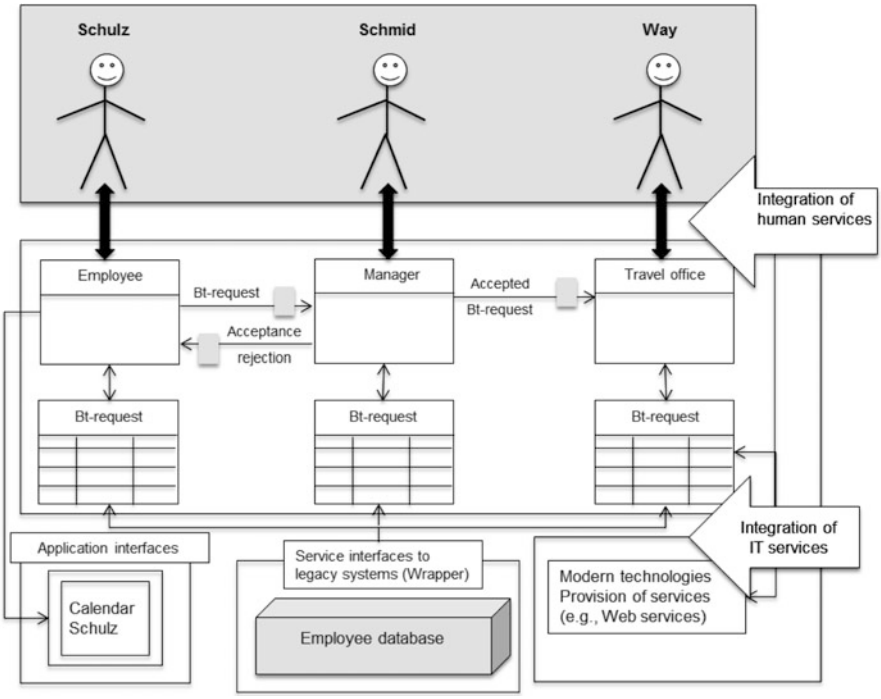

Fig. 10.5 Integration of services into the subjects of a process

abstraction (hiding function details), reuse (use by different consumers), and far-reaching autonomy (control over their own environment and resources) (cf. Erl 2008, pp. 86 et seq.)

- On the other hand, in the organization, as a rule legacy systems (applications) are in use which usually, especially for economic reasons (protection of investment, capital accumulation), cannot be easily converted in the short term into a modern, from the ground up designed service-oriented landscape (cf. e.g., Friend et al. 2008). Therefore, along the way the goal is to use mostly proven functionalities of existing systems, for instance, in that IT developers encase these functionalities using so-called wrapper programs (Legacy Wrapper). These separate functions from the monolithic structure and publish them as Web services, and so provide them as services in the sense mentioned above (cf. Mathas et al. 2008, pp. 111 ff.; Erl et al. 2008, p. 311; SOA Glossary 2011).

If access to a legacy application is preprocessed through a subject with wrapper properties, this handles the synchronous access to the functions of the application and provides the requester a usable asynchronous service. The consuming service is so less tightly coupled to the provider, compared to the case of self-contained, synchronous use of the function of the legacy system. This approach especially helps in meeting the demand for loose coupling of services.

In practice, often results of legacy applications need to be transferred to other legacy applications. This is achieved in subject orientation by sending the required data from the wrapper subject of the provider in form of a business object to the wrapper subject of the recipient. In this case, the subjects of legacy applications become service providers or service users, respectively.

- Finally, we can consider user interactions as services for subjects. Subjects use skills of their carriers, e.g., to enter data (such as, business trip data), to make decisions for the subsequent flow of the individual subjects and the overall flow of the process (e.g., approval or rejection of the business trip application).

In this way, human and IT services are bundled in a subject and integrated as a unit in a business process (see Fig. 10.5).

Implementing a service-oriented architecture for realizing S-BPM consequently leads to a distributed choreographic system. This enables IT resource optimization through flexible load sharing.

10.6.2 Service-Oriented S-BPM Architecture

SOA defines the logical architecture of the required service (bundles) for business process management. This business-oriented structure needs to be mapped to a corresponding physical infrastructure. Figure 10.6 shows an example of how this could be achieved. The dashed rectangles each represent different technical platforms.

The subject carriers use for their interactions within the process workplace computers, which are connected via proper networks to servers. These execute one or more subjects of the relevant business process, but possibly also other subjects of other processes. In the example, the subjects "employee" and "manager" run on the same physical system, while their business objects, e.g., for safety reasons, are located in separate environments, respectively. The subject "travel office" is located together with its business objects on a separate system. This could be due to the fact that for historical reasons the travel office has its own IT infrastructure, which is managed by an external partner. In addition, services required for communication among users or manipulating business objects were, e.g., for reasons of load balancing, distributed to separate systems, respectively.

Integration technologies need to be used for the interaction of solution components mapped to such a heterogeneous physical landscape. Figure 10.7 exemplifies a cross-selection of such technologies and the positions in the S-BPM architecture where they could be used. The numbers in the figure correspond to those in the subsequent explanation.

1. User interfaces are typically Web-based implementations. Here, different technologies, such as HTML, JavaScript, etc., can be used. For implementation, tools like Google Web Tool (GWT) and Flex (Adobe) are available. They offer

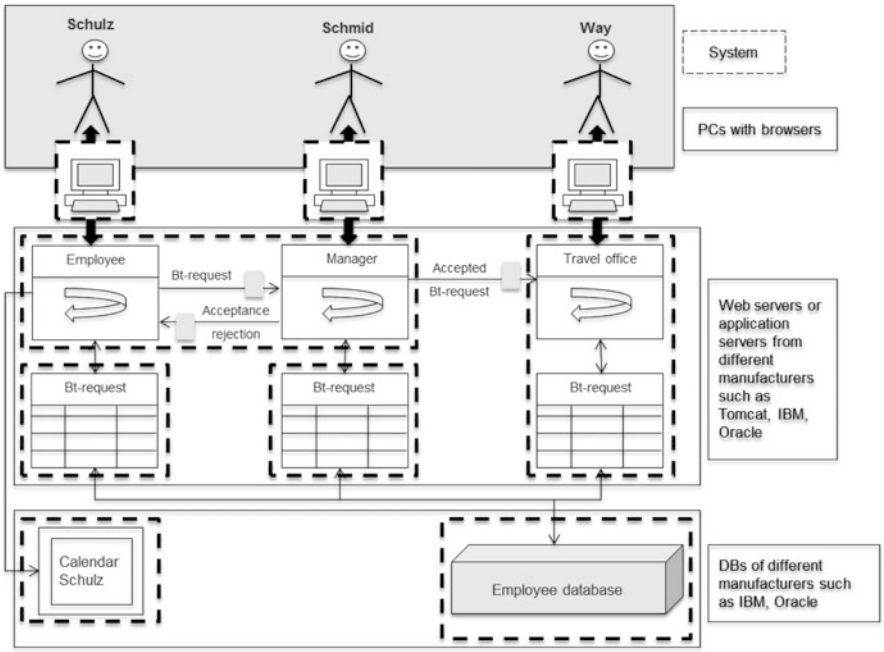

Fig. 10.6 Distribution of an S-BPM solution to multiple physical systems

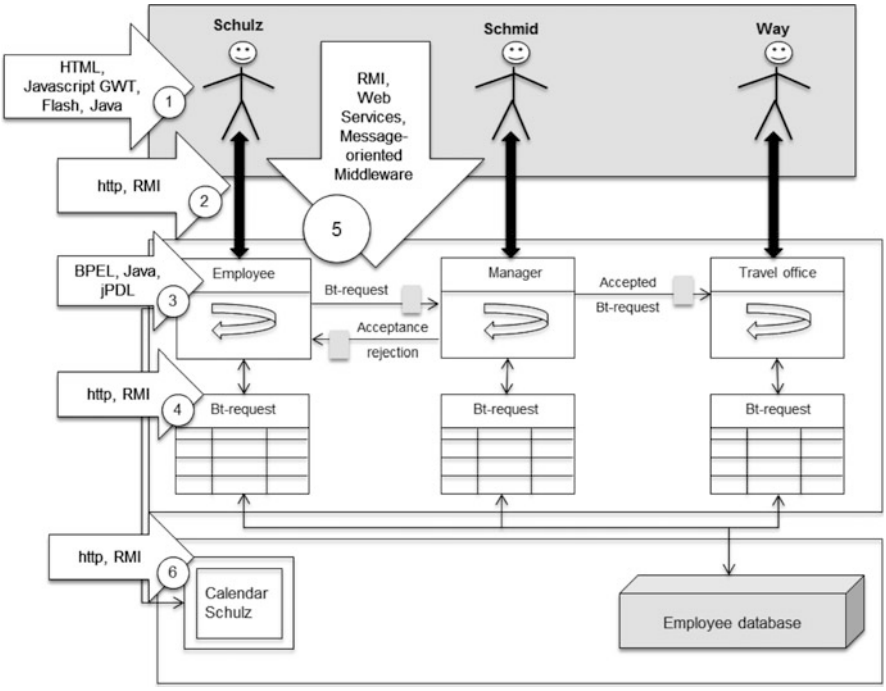

Fig. 10.7 Examples of integration technologies when implementing an S-BPM solution

off-the-shelf controls, e.g., selection boxes, selection lists, and table displays, which only have to be positioned by a developer at the appropriate location in the user interface. In order to structure a Web page, frames and other technologies, such as Master Pages in ASP.NET, are used. For the structuring of Web interfaces, portal technologies, such as portlets, are applied. They allow content presentation in an application- and user-oriented way when context stems from multiple sources. In this way, entries for process control in one portlet can access actual data for filling in a business object in a second, separate portlet. This is of particular advantage when business objects are manipulated by different form systems, such as Adobe Forms, eForm from IBM, or xForms. Portlets enable a high degree of flexibility when designing the user interface. The frameworks for assembling portlets to entire Web pages are supported by portal software offered by various manufacturers, such as IBM, Oracle, SAP, or, in the open source community, Liferay.

2. The communication of the users' PCs with the respective servers can be, depending on the realization of the user interface, via HTTP, or RMI implemented. The interaction of the users is controlled by the sequence control of the respective subject.
3. The flow control of the individual subjects and subject carriers, i.e., their behavior, can be separately implemented by different technologies, such as Java, BPEL, XPDL, or the like. This in turn determines which different runtime systems for each server are used. Web Application Servers already provide support for storing state information, for handling exceptions, or when restarting after a system crash.
4. For subject access to business objects, technologies such as Java, RMI, and Web services can be used.
5. For implementing the communication among the subjects, even across physical system boundaries when required, technologies such as RMI or Web services are used. The message exchange of subjects, including the input pool functionality, can be implemented, e.g., as a Web service. Compared to an RMI solution, in this case fewer problems with firewalls occur.
6. Databases can be connected directly via SQL commands, or when using Java via jDBC functions, to business objects. A flexible solution in this regard, based on Hibernate, is the hiding of vendor-specific features in SQL.

The type of technology used for coupling existing applications (legacy systems) strongly depends on the architecture in which they were developed. New applications usually provide an opportunity to trigger functions via Web service calls. In older systems, e.g., developed in Cobol, wrapper software may need to be used as an adapter, which allows calling COBOL programs from Java programs (cf. Herrmann et al. 2009).

The presented cross section of technologies demonstrates the flexibility in the implementation of S-BPM solutions, as well as the technological neutrality of the approach. Instead of using Java elements, a Microsoft.NET environment, for example, could also be used. The specific design can be completely aligned to the constraints and requirements of an organization. The subject-oriented architecture

helps in clearly spotting relevant areas with respect to technology and thus facilitates decision making regarding the implementation of BPM solutions.

The technological flexibility is especially demonstrated by the capability to provide different IT implementations for different organizational embeddings of a subject, which means for multiple subject carriers, within a specific process. This affects all aspects of process flow control, from manipulating business objects, to exchanging messages. For instance, an employee in the German headquarters may submit his business trip request via an SAP application, whereas employees of foreign subsidiaries accomplish this task via a Web interface. The flexible combination and integration of highly diverse technologies is of particular benefit in the case of inter- and cross-organizational processes.

References

Becker, J., Kugeler, M., Rosemann, M. (Hrsg.), Prozessmanagement, 6th edition, Berlin 2008.

Decker, G., Kopp, O., Leymann, F., Weske, M., BPEL4Chor: Extending BPEL for Modeling Choreographies, IEEE International Conference on Web Services, Salt Lake City, 2007, pp. 296-303.

Erl, T., SOA – Entwurfsprinzipien für Serviceorientierte Architektur, München, 2008.

Freund, J., Götzer, K., Vom Geschäftsprozess zum Workflow, München, 2008.

Gadatsch, A., Grundkurs Geschäftsprozess-Management, Wiesbaden 2010.

Herrmann, W., Java-Wrapper für COBOL-Funktionen, http://www.computerwoche.de/software/soa-bpm/1884724/index3.html, Download 11.03.2011. 2009

Krcmar, H., Informationsmanagement, 5th edition, Heidelberg 2010.

Mathas, C., SOA intern: Praxiswissen zu serviceorientierten IT-Systemen, München 2008.

Pahlke, I., Beck, R., Wolf, M., Enterprise-Mashup-Systeme als Plattform für situative Anwendungen, Wirtschaftsinformatik 52. Jg. (2010) 5, pp. 299–310.

Reinheimer, S., Lang, F., Purucker, J., Brügmann, H., 10 Antworten zu SOA. in: HMD – Praxis der Wirtschaftsinformatik, Heft 253, 2007, pp. 7–17.

Schmelzer, H., Sesselmann, W., Geschäftsprozessmanagement in der Praxis, 7th edition, München 2010.

SOA Glossary (2011) Definitions for Service-Oriented Computing Terms, http://www.soaglossary.com/legacy_wrapper.php, Download 11.03.2011

Vogler, P., Prozess- und Systemintegration, Wiesbaden 2006.

WfMC, The Workflow Reference Model, In: Lawrence, P. (Hrsg.), Workflow Handbook 1997, Chichester 1997.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution Non-commercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.